METHOD AND SYSTEM FOR USING EMULATION OBJECTS FOR DEVELOPING POINT OF SALE APPLICATIONS

FIELD OF THE INVENTION

The present invention relates to point of sale equipment and more particularly to a method and system for developing applications for point of sale equipment. The method and system allow for testing of the application that is simpler and more closely reflects performance of the application on the point of sale system. Thus, development of the application is facilitated.

BACKGROUND OF THE INVENTION

Point of sale (POS) equipment typically encompasses equipment which facilitates sales and which is physically located where the sales are made. For example, POS equipment may include cash registers that have some intelligence and a remote server to which the cash registers may be coupled. The POS equipment may be coupled to other specialized devices. These specialized devices include keyboards, scanners, scales, or other equipment usually residing in the same physical area as the POS equipment. For example, a cash register may be coupled to a scanner and a scale.

The POS equipment executes applications which use the specialized devices. For example, an application on the POS equipment may require input from the scale in order to calculate a price for an item that is sold based on the item's weight. Typically, drivers which also reside on the POS equipment control the specialized devices. In order to use the specialized device, the application communicates with the driver that controls the specialized device.

Typically, POS equipment has at most a limited environment for development of software. Consequently, the applications run on POS equipment are generally developed on a separate development system by a software developer. Once development is complete, the application can be adequately run on the POS system.

During conventional development of the application, the developer must ensure that the application will be capable of utilizing the specialized devices. However, the specialized devices are coupled to the POS equipment, rather than the development system. Thus, the developer must have access to POS equipment. Typically, this is accomplished by placing POS equipment in a lab, where many developers have access to the POS equipment and, therefore, the specialized devices.

Although placing the POS equipment in a lab allows applications to be tested, conventional development of the applications is difficult and time consuming. In order to test an application which has been written, a developer copies a portion of the application which has been written from the development system, takes the application to the lab, and downloads the application to the POS equipment. If the application does not function properly, the developer must change the code and repeat the entire process for the new version of the application. This is very time consuming. In addition, the developer will still be limited to working in proximity to the POS equipment because the developer must have access to the POS equipment for testing the application during development. This also makes the development process slower and more difficult. Although each developer can be provided with separate POS equipment, this may be expensive for the developer's employer.

Portions of code designed to account for the absence of the specialized devices on the development system can be added to the application. This allows the application with the

additional code to be tested on the development system. However, the application must subsequently be tested on the test system without these additional pieces of code once testing on the development system is completed. This subsequent testing shares many of the same problems as testing the application solely on the test equipment. Thus, development of applications for POS equipment using conventional mechanisms is inefficient and time consuming.

Accordingly, what is needed is a system and method for facilitating development of applications to be used on POS equipment. The present invention addresses such a need.

SUMMARY OF THE INVENTION

The present invention provides a method and system for developing an application. The application is for use with point of sale equipment having a device. The application is capable of utilizing the device when the application is executed on the point of sale equipment. The method and system comprise providing an emulation module corresponding to the device. The method and system further comprise ensuring that the application will utilize the emulation module when the application is executed on the development system. Thus, when the application is executed on the system, the emulation module and the application emulate the interaction between the application and the device that occurs when the application is executed on the point of sale equipment.

According to the system and method disclosed herein, the present invention allows the application to be more easily and accurately tested, thereby increasing facilitating development of the application.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is a block diagram of a system including point of sale equipment.

Figure 1B is a block diagram of a conventional system for developing applications to be run on point of sale equipment.

Figure 2 is a flow chart depicting a conventional method for developing applications to be run on point of sale equipment.

Figure 3 is a block diagram of a development system in accordance with the present invention.

Figure 4 is a flow chart depicting a method for providing a development environment in accordance with the present invention.

Figure 5 is a flow chart depicting a method for testing an application in accordance with the present invention.

Figure 6 is a flow chart depicting the interaction between the emulation object and the application in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to an improvement in development of applications for point of sale equipment. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features

described herein.

Figure 1A is a block diagram of a system 10 including point of sale equipment that uses specialized devices. The system 10 includes a server 12 coupled with three stations 20, 30, and 40. The stations 20, 30, and 40 include POS equipment which is used when and where the purchase is actually made. For example, the stations 20, 30, and 40 may include cash registers having additional intelligence. The stations 20, 30, and 40 are each coupled to specialized devices. The specialized devices communicate with the stations 20, 30, and 40, and provide special functions to each of stations. Particular specialized devices are depicted in Figure 1A. The station A is coupled to a scanner A 22 and a scale A 24. The station B is coupled to a scanner B 32. The station C 40 is coupled to a scale C 42, a scanner C 44, and a keyboard C 46. Typically, drivers 23, 25, 33, 43, 45, and 47 control the specialized devices 22, 24, 32, 42, 44, and 46, respectively. Thus, the application 15 interfaces with the drivers 23, 25, 33, 43, 45, and 47 to use the devices 22, 24, 32, 42, 44, and 46, respectively.

The stations 20, 30, and 40 include an application 15. The application 15 is run on the stations 20, 30, and 40. Often, the application 15 may use some or all of the specialized devices 22, 24, 32, 42, 44, and 46. Thus, when ~~such~~ the application 15 is developed, the developer must ensure that the application 15 functions with the specialized devices 22, 24, 32, 42, 44, and 46. Typically, this means that the developer must ensure that the application can communicate with the drivers 23, 25, 33, 43, 45, and 47 for the specialized devices 22, 24, 32, 42, 44, and 46, respectively, and control the devices 22, 24, 32, 42, 44, and 46, respectively, through the drivers 23, 25, 33, 43, 45, and 47, respectively. Note that although the same application 15 is shown on all stations 20, 30, and 40, nothing prevents the stations 20, 30, and 40 from running different applications.

Figure 1B depicts a conventional development system 50 for developing an application for use with a system 10 including POS equipment. Typically, POS equipment such as the stations 20, 30, and 40 have very limited development tools, if any. Consequently, development typically occurs using the separate conventional development system 50. The conventional development system 50 includes a computer 51 and a display 56. The computer 51 includes an operating system 52, development tools 54, and the application 15 currently being developed. The development tools 54 and operating system 52 create a development environment that is significantly more sophisticated ~~that~~ then may be available in the system 10.

Figure 2 depicts a conventional method 60 for developing an application 15 for use with a system 10 including POS equipment. The method 60 is used in conjunction with the conventional development system 50. The code for the application 15 is written or rewritten, via step 62. The code for the application 15 is then compiled on the development system, via step 64. Although the application 15 has been compiled in step 64, the application 15 may still be unable to function adequately on the system 10 having POS equipment. In order to function adequately, the application 15 must be able to correctly use the specialized devices 22, 24, 32, 42, 44, and 46. Thus, once the code for the application 15 is compiled in step 64, the application 15 must be tested to ensure that the application 15 can use with the specialized devices 22, 24, 32, 42, 44, and 46. The development system 50 does not include the specialized devices 22, 24, 32, 42, 44, and 46. Consequently, the application 15' must be moved to a test system (not shown) having the specialized devices 22, 24, 32, 42, 44, and 46 for testing.

To test the application 15, the compiled application 15 is copied, for example to a floppy disk, via step 66. The copy of the application 15 is then transferred to a test system, via step 68. The test system has one or more specialized devices that the application may use. For

example, one test system could be similar to the station C 40 and include scale C 42, scanner C 44, and keyboard C 46. The application 15 is then run on the test system, via step 70. It is then determined whether the application 15 functions adequately on the test system, via step 72. Step 72 includes determining whether the application 15 can use the specialized devices coupled to the test system. If the application 15 functions adequately, then the developer is finished. However, if the application 15 does not function adequately on the test system, then steps 62 through 72 are repeated until the performance of the application is satisfactory.

Although the method 60 functions, one of ordinary skill in the art will realize that the method 60 has several drawbacks. The combination of the step 66 of copying the application and the step 68 of transferring the application 15 to the test system is tedious and time consuming. Step 68 typically includes physically transferring the application 15 to the test system and downloading the application 15 to the test system so that the application 15 can be run on the test system. This is a tedious process. In addition, if it is determined in step 72 that the application does not adequately function with the test system, then the whole method 60, including copying the application 15 and transferring the application 15 to the test system, must be repeated. The application could be used on a variety of systems 10 that use several different platforms. If the application 15 is not platform independent, the method 60 must also be repeated for each platform on which the application may be run. This is because the drivers for specialized devices 22, 24, 32, 42, 44, and 46 may be different on each platform. Thus, development of the application is further slowed.

Moreover, the test system must be provided to allow the developer to test the application 15. Typically, a lab including one or more test systems is provided for use by several developers. However, if there are many developers using the lab, a developer may have

to schedule time on the test system. In addition, because the developer must repeatedly use the lab during development, the developer cannot efficiently develop the application 15 in another location, such as the developer's home. Although providing each developer with at least one test system would solve some of these problems, the cost may be prohibitive and would leave the other problems unsolved.

Finally, limited testing of the application 15 may be performed on the conventional development system 50. However, in order to do so, additional code designed to account for the absence of the specialized devices on the conventional development system 50 must be added to the application. The application 15' plus the additional code is then run on the conventional development system 50, rewritten to overcome inadequacies, and run again until the application 15' plus the additional code can run adequately on the conventional development system. Once testing on the conventional development system is complete, this additional code is removed from the application 15'. The application 15' must then be tested without the additional code on the test system. Typically, the testing of the application 15' on the test system is relatively extensive. Thus, the method 60 is still used for the application 15'. Thus, many drawbacks of the method 60 are still present.

The present invention provides for a method and system for developing an application for use with point of sale equipment having a device. The application is capable of utilizing the device when the application is executed on the point of sale equipment. The method and system comprise providing an emulation module corresponding to the device. The method and system further comprise ensuring that the application will utilize the emulation module when the application is executed on the development system. When the application is run on the system, the emulation module and the application emulate the interaction between the

application and the device that occurs when the application is executed on the point of sale equipment.

The present invention will also be described in terms of particular devices and a particular language. However, one of ordinary skill in the art will readily recognize that this method and system will operate effectively for other devices and for other languages. Moreover, the present invention is described in the context of modules used in object oriented programming. However, nothing prevents the use of the present invention in conjunction with other types of programming. Although the present invention finds particular utility in a platform independent context, the present invention need not be platform independent.

To more particularly illustrate the method and system in accordance with the present invention, refer now to Figure 3 depicting a block diagram of one embodiment of such a development system 100. The system 100 is for use in developing applications for the system 10 that includes POS equipment. The system 100 a computer 101 coupled to a display 120. The computer 101 includes an operating system 102, development tools 104, an application 15' that is currently being developed, and emulation modules 110, 112, and 114. The development tools 104 are for use in facilitating developing the application 15'. The combination of the development tools 104 and the operating system 102 provide a development environment that is more sophisticated than in the system 100. The application 15' is analogous to the application 15. Thus, the application 15' is being developed for use on a system 10 including point of sale equipment 20, 30, and 40 having specialized devices 22, 24, 32, 42, 44, and 46. The emulation modules 110, 112, and 114 correspond to the devices (not shown in Figure 3) which the application 15' will use when executing on the system for which the application 15' is being developed. For the purposes of discussion it is presumed that the emulation modules

110, 112, and 114 are for emulating the devices 42, 44, and 46 on the station C 40. Thus, when the application is run on the development system 100, the emulation objects 110, 112, and 114 make it appear to the application 15' as though the devices 42, 44, and 46, respectively, are present and being used. In a preferred embodiment, the emulation modules 110, 112, and 114 emulate the devices 42, 44, and 46 and the corresponding drivers 43, 45, and 47.

In a preferred embodiment, modules 110, 112, and 114 are platform independent and object oriented. Consequently, they will be referred to as emulation objects 110, 112, and 114. The emulation objects 110, 112, and 114 can be used to emulate the interaction between the application 15' and specialized devices 42, 44, and 46 which the application 15' may use when run on the real point of sale equipment. The emulation objects 110, 112, and 114 perform this function when the application 15' is run on the system 100. In one embodiment, the emulation objects 110, 112, and 114 are written in JAVA and emulate the drivers 43, 45, and 46, respectively, and the scale C 42, the scanner C 44, and the keyboard C 46, respectively. The emulation objects 110, 112, and 114 make it appear to the application 15' that the application is actually utilizing the specialized devices 42, 44, and 46. In other words, when the application 15' is run on the development system 100, the application will encounter the emulation objects 110, 112, and 114 that behave as though the application is communicating directly with the devices 42, 44, and 46.

Figure 4 depicts a method 150 for providing a development system in accordance with the present invention. The emulation objects 110, 112, and 114 are provided on the development system 100, via step 160. Thus, the emulation objects 110, 112, and 114 are made available to the developer when developing the application 15'. It is then ensured that when the application 15' is executed on the development system 100, the application 15' will

utilize the emulation objects 110, 112, and 114, via step 170. In a preferred embodiment, step 170 includes ensuring that the application 15' will encounter the emulation objects 110, 112, and 114 when the application 15' is run on the development system 100. Once the application 15' encounters the emulation objects 110, 112, and 114, the application 15' utilizes the emulation objects 110, 112, and 114 as though the application was using the devices 42, 44, and 46 to which the emulation objects 110, 112, and 114 correspond.

The method 150 and the development system 100 in accordance with the present invention ensure that the developer can more accurately test the application 15' on the development system 100, rather than using a separate test system. The method 150 and development system 100 also ensure that the developer can test the application 15' on the development system 100 without placing additional code in the application 15'. Furthermore, running the application 15' on the development system 100 will give the developer a more accurate indication of the behavior of the application 15' on the system 10 than adding additional code to the application 15'.

Figure 5 depicts a more detailed flow chart of a preferred embodiment of a method 200 in accordance with the present invention for developing the application 15'. Referring to Figures 3 and 5, the method 200 is used in conjunction with the system 100. It is ensured that the emulation objects 110, 112, and 114 are on the development system 100, via step 210. Step 210 includes providing the emulation objects 110, 112, and 114 in JAVA. Thus, the emulation objects 110, 112, and 114 are platform independent.

Via step 220, the emulation objects 110, 112, and 114 are then placed higher in the class path than the objects which will provide communcation with the drivers 43, 45, and 47, respectively, for the devices 42, 44, and 46, respectively, which the application 15' will use.

Step 220 is analogous to the step 170 of the method 150. In JAVA, applications access devices 42, 44, and 46 based on the class of each device 42, 44, and 46. A class path defines the order of the locations in which a JAVA application will look to find object classes. An object class which is higher in the class path will be located before and accessed in lieu of an object class of the same name that is lower in the class path. Because the emulation objects 110, 112, and 114 are placed higher in the class path than object classes of the same name which provide communication with the real device driver 43, 45, and 47, respectively, the application 15' will locate the emulation objects 110, 112, or 114 first when the application 15' is being executed on the development system 100. Thus, the application 15' will use the emulation objects 110, 112, or 114 when the application 15' is executed on the development system 100.

The code for the application 15' is written or modified and compiled, via step 230. To test the application 15', the application 15' is run on the development system 100, via step 240. Because steps 210 and 220 have been carried out, the application 15' will use the emulation objects 110, 112, and 114 in step 230. It is then determined if the application 15' functions adequately, via step 250. Step 250 preferably includes determining whether the application 15' can use the emulation objects 110, 112, and 114 properly. If not, then steps 230, 240, and 250 are reiterated until the application 15' functions adequately.

Once the application 15' functions adequately on the development system, a final debug is performed, via step 260. Step 260 includes running the application 15' on a test system (not shown) which actually includes the devices 42, 44, and 46 that the application 15' may use. Thus, the application 15' is preferably run on a test system (not shown) which is includes the station C 40 and the devices 42, 44, and 46. Because the emulation objects 110, 112, and 114 are not on either the test system or the actual system 10, the application 15' will

utilize the devices 42, 44, and 46 when run on the test system or the station C 40. The final debug performed in step 260 accounts for any differences, such as minor timing differences, between using the actual hardware for the devices 42, 44, and 46, and using the emulation objects 110, 112, and 114. However, these differences may be minimal, making the final debug step 260 relatively simple. Development of the application 15' is then complete.

Figure 6 depicts a flow chart of a method 300 indicating a portion of the interaction between the application 15'and the emulation objects 110, 112, and 114 while the application 15' is run on the development system 100. The application 15' enables the emulation objects 110, 112, or 114, via step 302. The emulation object then provides a window (not shown) on the display 120 of the development system 100, via step 304. From the window, the user can provide input to the application 15'. The emulation objects 110, 112, or 114 also respond to a device specific command, via step 306. The emulation objects 110, 112, or 114 provide the response expected from the corresponding device 42, 44, or 46, respectively, in step 306. The application 15' may expect input from the devices 42, 44, or 46 corresponding to the emulation objects 110, 112, or 114, respectively. If so, then the developer provides the input to the window, via step 308. The emulation object 110, 112, or 114 then provides the input to the application 15' in step 310. The emulation object 110, 112, or 114 provides the input in the form expected from the drivers 43, 45, or 47 of the devices 42, 44, or 46. Referring back to Figure 5, the developer can then determine in step 250 if the application 15' has adequately responded to the information provided from the emulation objects 110, 112, or 114. For example, when the application is being run on the system 10, the scanner 40 may provide a stock number to the application 15' so that the application 15' can calculate a price. The developer can provide the stock number in step 308. The emulation object 112 provides the

stock number to the application 15' in the form expected from the driver 45 of the scanner 44. The developer can then determine if the application 15' has calculated the price correctly. If not, then the developer can modify and retest the application on the development system 100 using the method 200.

Because the emulation objects 110, 112, and 114 are used, the developer can obtain an accurate picture of how the application 15' will function on point of sale equipment for which the application 15' is being developed. As a result, the tedious and repeated transfer of the application to the equipment is avoided. When the emulation objects 110, 112, and 114 and the application 15' are platform independent, the same code is executed on the real system 10 and the development system 100. Therefore, a very complete picture of the application's behavior is obtained by testing the application on the development system 100. Even when the emulation objects 110, 112, and 114 and the application 15' are not platform independent, the same source code for the application 15' may be used. However, sets of emulation objects 110, 112, and 114 specific to the platform used by the system 10 are preferably used. Thus, emulation objects 110, 112, and 114 are preferably provided for each platform when the emulation objects 110, 112, and 114 and the application 15' are platform dependent. For example, if the application 15' is written in C, the same source code could be used. The source code ~~could~~ would be compiled and linked for each platform with which the code is to be used. The linked code would then be used with emulation objects 110, 112, and 114 specific to each platform. However, in a preferred embodiment, the emulation objects 110, 112, and 114 are platform independent.

In either the platform independent or the platform dependent case, additional code need not be added to the application 15'. Furthermore, testing on the development system 100

provides a more accurate picture of the behavior of the application 15' than testing the application 15 with additional code on the conventional development system 50. In addition, the class path determines whether the emulation objects 110, 112, and 114 or the devices 42, 44, and 46, respectively, will be used. It is, therefore, easy for a developer to determine whether the application 15' is attempting to use the emulation objects 110, 112, and 114, or the devices 42, 44, and 46, respectively.

The developer is also free to develop the application 15' on any development system 100 having the emulation objects 110, 112, and 114. Thus, the developer need not remain near a lab having separate test equipment in it or incur the expense of having separate test equipment at each location the developer desires to work. In addition, the final debug using the separate test equipment is significantly simpler. Therefore, development of the application 15' is made simpler, faster, and more efficient.

A method and system has been disclosed which allows a developer to more efficiently develop applications executed by point of sale equipment coupled to devices used by the application. Software written according to the present invention can be stored in some form of computer-readable medium, such as memory or CD-ROM, or can be transmitted over a network, and executed by a processor.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.